UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
TYLER DIVISION

FENNER INVESTMENTS, LTD., Plaintiff, v. 3COM CORPORATION, FOUNDRY NETWORKS, INC., EXTREME NETWORKS, INC., NETGEAR INC., ZYXEL COMMUNICATIONS, INC., D-LINK SYSTEMS, INC., SMC NETWORKS, INC., TELLABS, INC., TELLABS NORTH AMERICA, INC., AND ENTERASYS NETWORKS, INC., Defendants.	Case No.: 6:08-CV-00061 – (LED) JOINT CLAIM CONSTRUCTION AND PREHEARING STATEMENT PURSUANT TO PATENT LOCAL RULE 4-3

Plaintiff Fenner Investments, Ltd. (“Fenner”) and Defendants 3Com Corporation, Foundry Networks, Inc., Extreme Networks, Inc., NETGEAR, Inc., ZyXEL Communications, Inc., D-Link Systems, Inc., SMC Networks, Inc., Tellabs, Inc., Tellabs North America, Inc., and Enterasys Networks, Inc. (“Defendants”), by and through their respective counsel, hereby respectfully submit the following Joint Claim Construction and Prehearing Statement pursuant to P.R. 4-3 of the United States District Court for the Eastern District of Texas.

The parties exchanged proposed terms and claim elements for construction pursuant to P.R. 4-1. The parties thereafter exchanged proposed constructions for each term and claim element pursuant to P.R. 4-2 and conducted a meet-and-confer conference regarding the proposed terms and claim elements. The parties expressly reserve their rights to propose

constructions of additional terms, phrases or clauses in the asserted patents at a later time. In addition, the parties expressly reserve their rights to supplement or amend the proposed construction and other positions set forth herein.

Table of Contents

I. P.R. 4-3(a): CLAIM TERMS, PHRASES, OR CLAUSES ON WHICH THE PARTIES AGREE

Pursuant to P.R. 4-3(a), the parties provide constructions for those claim terms, phrases, or clauses on which they agree:

Claim term	*Joint construction*
MAC source address found in the '906 patent, claim[s]: 9, 10, 19, 20	MAC address (as construed herein) of origin
MAC destination address found in the '906 patent, claim[s]: 9, 10, 19, 20	MAC address (as construed herein) to which something is sent
source filtering information / source address filtering information found in the '224 patent, claim[s]: 3, 8 found in the '906 patent, claim[s]: 9, 19	information used to determine whether to filter a packet based on the packet's source address
stored protection record found in the '906 patent, claim[s]: 10, 20	record containing information used to determine whether to filter a packet based on the packet's source address

II. P.R. 4-3(b): DISPUTED CLAIM TERMS, PHRASES AND CLAUSES

Pursuant to P.R. 4-3(b), the parties identify the following claim terms, phrases, or clauses on which they disagree, and each submits its proposed constructions, along with an identification of intrinsic and extrinsic evidence in support thereof to the Court:

A. Proposed Constructions of Each Disputed Claim Term, Phrase, or Clause

1. U.S. Pat. No. 5,842,224

*Claim language of the '224 Patent (disputed terms in **bold**)*	*Plaintiff's proposed construction and supporting evidence*	*Defendants' proposed construction and supporting evidence*
logical address found in claim[s]: 3	An address assigned within a computer network; examples include IP addresses. <u>INTRINSIC EVIDENCE</u>: Column 11, lines 1-4; Column 21, lines 37-40; Column 23, lines 45-48 <u>EXTRINSIC EVIDENCE</u>: Wiley Electrical and Electronics Engineering Dictionary, p. 433	a fixed, unique, and unchanging identifier of a connection to the internet represented by a series of numbers that has no internal structure to suggest network connection location <u>INTRINSIC EVIDENCE</u>: *'224 patent specification*: Col. 2:37-46 ("The present system overcomes the disadvantages of the prior art by simply assigning a fixed, unique and unchanging identification code to both host A and host B. As host B enters into a new network access system, it transmits its identification code to the nearest node and all of the nodes interconnecting all of the disparate networks each store, with the unique identification code of host B, the address of those nodes which can communicate with host B so that a path can be completed through the nodes between host A and host B.") Col. 11:1-18 ("The novel system of the present invention modifies FIG. 1 to provide an Internet routing table that uses a flat logical address structure to provide fast and efficient route processing of both multicast and unicast message traffic. In the present system, the physical address structure is removed from the design and operation of the Internet routing by treating the message addresses as a symbol string without predetermined internal structure and processing them as if they are a unique identification code representing the host. This approach is made possible by employing an arithmetic code compression technique as a hashing function for the routing table access method. By managing and manipulating logical network addresses within the system, mobile end systems can keep the same network

*Claim language of the '224 Patent (disputed terms in **bold**)*	*Plaintiff's proposed construction and supporting evidence*	*Defendants' proposed construction and supporting evidence*
		identification code (not physical address) as they move from communication network to communication network. Similarly, group or multicast addresses may be allocated without regard to their physical network connection.") Col. 5:30-39 ("The present invention provides a very fast, automatically expandable, source filtered Internet routing scheme totally independent of the internal logical or physical structure of the network addresses in the message format that it is routing. Addresses are just unique identification numbers represented by a string of symbols of known length. Each Internet router learns the location of these numbers within the network from the Internet protocol traffic, from the source addresses of the packets it receives, and from a network management protocol.") Col. 4:38-44 ("The present invention overcomes the disadvantages of the prior art by considering a flat, as opposed to hierarchical, logical routing address space with unique identifiers assigned to each transmitter and receiver to vastly simplify the modern communication problems of addressing highly mobile end-systems which are simultaneously connected to multiple communication paths and employ multicast message traffic.") Col. 8:55-62 ("If aircraft 10 desires to contact ship 12, it simply transmits a message format including its own unique code and the unique identification code for ship 12 to the nearest system 14. The receiving system 14 sends the message to node 18 which checks its memory tables to determine if it has stored the address of the last node (26 or 32) communicating with ship 12. If not, it stores the unique identification code of aircraft 10.") Col. 10:35-39 ("The problem with such vehicle movement with the prior art system, as stated, is that each of the communication systems 1-5 are different networks and may use different types of media access protocols for operation which

*Claim language of the '224 Patent (disputed terms in **bold**)*	*Plaintiff's proposed construction and supporting evidence*	*Defendants' proposed construction and supporting evidence*
		require the network address of the moving vehicle to be changed.") Col. 10:52-58 ("As stated, in the prior art, the aircraft 10 must have assigned to it a code representing its physical address with respect to communication system 1. Physical addresses are associated with interface hardware. Thus, moving the hardware interface to a new machine or replacing a hardware interface that has failed changes the physical address of a particular host.") Col. 10:52-67 ("As stated, in the prior art, the aircraft 10 must have assigned to it a code representing its physical address with respect to communication system 1. Physical addresses are associated with interface hardware. Thus, moving the hardware interface to a new machine or replacing a hardware interface that has failed changes the physical address of a particular host. In like manner, as the aircraft moves from system 1 to system 2 in FIG. 1 . . . the coding of the physical address of aircraft 10 must be changed to meet the standards of system 2. This means that if ship 12 attempts to communicate with aircraft 10 using the physical address at the last known address location in system 1, it cannot locate aircraft 10 without a new location code because aircraft 10 has moved into a new communication system network and has changed its physical address code.") *'224 patent prosecution history*: June 9, 1997 Amendment, pages 11-12 ("New claims 46-56 are directed to a method and apparatus for filtering data packets based on a source's logical address, not its physical address. . . . [T]here is a physical or media address in which a data packet is encapsulated, and a logical address for the source in the data packet which is independent of the source's physical or media address. During routing at each node interconnecting networks, the physical address layer is stripped off and replaced

*Claim language of the '224 Patent (disputed terms in **bold**)*	*Plaintiff's proposed construction and supporting evidence*	*Defendants' proposed construction and supporting evidence*
		with the physical address of the node to which it is next forwarded. . . . One benefit of filtering based on the logically (sic) address of the source is that it is device independent. It does not matter what network or device the source then currently resides. Thus, source filtering may take place in an internet environment or with mobile users.") Appeal Brief, page 7 ("Claims 46, 51, and 55 are directed to a method and apparatus for filtering data packets based on a source's logical address, not its physical address. . . . One benefit of filtering based on the logically (sic) address of the source is that it is device independent. It does not matter what network or device the source then currently resides or has accessed. Thus, source filtering may take place in the Internet environment, in which the physical addresses of the source are stripped from a packet by an intermediary node, or with a mobile source which may access the data networks through different network media, and thus may not be assigned the same physical address during each access.")
source address found in claim[s]: 8	address of origin INTRINSIC EVIDENCE: Figure 4; Column 5, lines 35-39; Column 7, lines 14-21; Column 12, lines 1-3; Column 13, lines 1-6; Column 14, lines 2-25 EXTRINSIC EVIDENCE: Merriam-Webster Dictionary (definition of "source")	a fixed, unique, and unchanging identifier that has no internal structure to suggest network connection location and that is assigned to the host sending the data packet INTRINSIC EVIDENCE: Please refer to the intrinsic evidence reference above for the claim term "logical address."
source address for logically identifying the sender of the data packet found in claim[s]: 8	a source address (as construed herein) for logically identifying the sender of the data packet INTRINSIC EVIDENCE: Figure 4;	a fixed, unique, and unchanging identifier that has no internal structure to suggest network connection location and that is assigned to the host sending the data packet

*Claim language of the '224 Patent (disputed terms in **bold**)*	*Plaintiff's proposed construction and supporting evidence*	*Defendants' proposed construction and supporting evidence*
	Column 5, lines 35-39; Column 7, lines 14-21; Column 12, lines 1-3; Column 13, lines 1-6; Column 14, lines 2-25 EXTRINSIC EVIDENCE: Merriam-Webster Dictionary (definition of "source")	INTRINSIC EVIDENCE: Please refer to the intrinsic evidence reference above for the claim term "logical address."
destination address for logically identifying a recipient of the data packet found in claim[s]: 12	the address where something is sent that logically identifies a recipient of the data packet INTRINSIC EVIDENCE: Figure 4; Column 7, lines 14-21; Column 11, lines 48-59; Column 12, lines 1-7	a fixed, unique, and unchanging identifier that has no internal structure to suggest network connection location and that is assigned to the host receiving the data packet INTRINSIC EVIDENCE: Please refer to the intrinsic evidence reference above for the claim term "logical address."
looking up, in a directory table stored at the node, source filtering information associated with the first logical address found in claim[s]: 3	looking up, in a directory table stored at the node, source filtering information (as construed herein) associated with the first logical address (as construed herein) INTRINSIC EVIDENCE: Column 13, lines 1-6; Column 11, lines 1-4; Column 21, lines 37-40; Column 23, lines 45-48 EXTRINSIC EVIDENCE: Wiley Electrical and Electronics Engineering Dictionary, p. 433	retrieving source filtering information (as construed herein) contained in a record identified by a unique value created by arithmetically compressing, as distinct from hashing, the first logical address INTRINSIC EVIDENCE: *'224 specification*: Figures 2-4 Abstract ("To provide for fast access times with very large key fields, an associative memory utilizes a location addressable memory and look up tables to generate from a key an address in memory storing an associated record. The look up tables, stored in a memory, are constructed with the aid of arithmetic data compression methods to create a near perfect hashing of the keys. For encoding into the look up table, keys are divided into a string of symbols. Each symbol is assigned an index value, such that a modulo sum of index values for symbols of a particular key is a unique value that is used as an address to the memory storing the record associated

*Claim language of the ‘224 Patent (disputed terms in **bold**)*	*Plaintiff’s proposed construction and supporting evidence*	*Defendants’ proposed construction and supporting evidence*
		with that key.”) Col. 1:13-15 (“The present invention relates to associative memory systems, and more particularly to associative memory systems for handling large key set and spaced.”) Col. 4:45-49 (“Further, the present invention employs a reversible arithmetic code compression technique to reduce the logical network address of up to 128 bits to a unique integer value which preserves any hierarchical ordering of the network address.”) Col. 4:62-65 (“Arithmetic coding, when applied to addresses as known length keys, provides several advantages for table look-up when the addresses are known or can be learned in advance as they are in communications applications.”) Col. 5:12-29 (“Secondly, arithmetic coding can be constructed to operate on each symbol position in the address field as it arrives, allowing processing to begin as soon as the first address symbol arrives. Thirdly, arithmetic coding preserves the hierarchical (left to right precedence) of the IS0 addresses being encoded. This is desirable if an Internet router only has knowledge of the network address but the Internet header carries the full destination address of a succeeding system node. Finally, a constant known set of computations is required for each symbol of the address field independent of the number of address symbols or the number of active Internet addresses. These features make the arithmetic coding used herein an ideal candidate for the routing table directory structure that is independent of a location address in a router, gate way or end-system.”) Col. 5:34-39 (“Addresses are just unique identification numbers represented by a

*Claim language of the '224 Patent (disputed terms in **bold**)*	*Plaintiff's proposed construction and supporting evidence*	*Defendants' proposed construction and supporting evidence*
		string of symbols of known length. Each Internet router learns the location of these numbers within the network from the Internet protocol traffic, from the source addresses of the packets it receives, and from a network management protocol.") Col. 5:65- Col. 6:1 ("The present invention combines arithmetic coding with dynamic hashing to provide a very high speed method and system for detecting the 48 bit physical addresses in a Media Access Controller (MAC). ") Col 6: 37-56 ("Another aspect of the invention is an apparatus and method for implementing a routing table directory to provide for fast access times to look up routing information. This apparatus is an application of a novel associative memory utilizing arithmetic coding to associate a key presented to the memory with a record stored in the memory, but has a very-wide range of application in many different types of data processing systems. The associative memory includes an index table stored in memory and a record memory for storing the records of data. The index table is constructed such that each symbol of a key, a key being divided into a string of symbols and each symbol being defined by its position within the key and its value, addresses an index value in the index table memory. These index values are assigned such that the sum of index values for a given key is a unique value that is used to address the record memory. Several methods and apparatus are disclosed the permit random assignment of index values to new keys as they are presented, as well as for keys that are presented in sorted order for addition to the memory.") Col. 11:1-12 "[T]o provide an Internet routing table that uses a flat logical address structure to provide fast and efficient route processing of both multicast and unicast message traffic. In the present system, the physical address structure is removed from the design and operation of the Internet routing by treating the message addresses as a symbol string without predetermined

*Claim language of the '224 Patent (disputed terms in **bold**)*	*Plaintiff's proposed construction and supporting evidence*	*Defendants' proposed construction and supporting evidence*
		internal structure and processing them as if they are a unique identification code representing the host. This approach is made possible by employing an arithmetic code compression technique as a hashing function for the routing table access method." Col. 13:44-65 ("The address detection logic examines both the destination and source address fields represented by the octets shifted into buffer 48 and buffer 50. Six octets are in each buffer. When the twelve octets are all stored, each octet is used as an address into a 256 element index table for that address octet position. This requires six destination index tables 66 and six source index tables 68. The output of these tables (the contents of the location addressed by each octet) is then arithmetically combined in combiners 70 and 72. One method of arithmetically combining these outputs adds the six outputs of the source index table 68 to compute the source index 74. It also adds the six destination table 66 outputs to compute the route index 76. The source index 74 is used as the address into the source protect table 78 and the output of that location is the source protection record 80 which is coupled to the routing logic 82. Similarly, the route index 76 is used as the address of a location in the destination routing table 84 and the contents of that location is coupled to route record 86. The outputs of the protect record 80 and route record 86 are used by the routing logic 56 in a well-known manner to determine which destination MAC is to receive the message.") Col. 14:56-65 ("In FIG. 2, the address detection logic employs separate tables and arithmetic processing elements for both the source and destination address detection. While this approach allows the arithmetic processing and record table access to be relatively slow the slower elements are not sufficiently economical in price to be cost effective. Neither does the circuit of FIG. 2 utilize the fact that because the data octets arrive sequentially, they could be

*Claim language of the '224 Patent (disputed terms in **bold**)*	*Plaintiff's proposed construction and supporting evidence*	*Defendants' proposed construction and supporting evidence*
		processed through the index look-up table and partial arithmetic computed each octet time. ") Col. 14:66-col. 15:42 ("FIG. 3 is a circuit diagram of an alternate logic layout for serial processing of the incoming data by a switch 96 which is similar to switch 34. The data octets arrive sequentially and FIG. 3 discloses a logic layout which uses one bank of index table memory 98, one bank of address record memory 100 and one arithmetic computation unit 102 to accomplish both source and destination address detection. In this approach, the octet data bits are coupled serially into octet register 104 and are used as the low order address bits into the index table 98. A byte counter 106 which counts the address octets from one to twelve as they arrive in the octet register 104 is used as the high order address bits into the index table 98. From byte count one to six, the arithmetic unit 102 partially computes the final index with each output from the index table. After byte count six, the computation for the destination address mask index is complete and transferred to the index buffer 108. The arithmetic unit 102 is then reset and the six octets of source address are computed. By the time the source protection record index 110 has been computed, the data in destination route record 112 has been loaded into its output buffer on line 114. The source protect record 110 is then accessed from a second bank of the mask memory 100 using the count twelve signal on line 116 as the high order address bit. This sequential detection approach shown in FIG. 3 places special performance requirements on the index table memory and each reiteration of the arithmetic computation. That is, the access to the index table 98 and the partial computation with the table output each must be complete in less than one octet time. However, the computation is delayed one octet clock time behind the table access. Each of these timing requirements is within the available speeds of commercially available VLSI computer memory and arithmetic components.

*Claim language of the '224 Patent (disputed terms in **bold**)*	*Plaintiff's proposed construction and supporting evidence*	*Defendants' proposed construction and supporting evidence*
		Current DRAM memories regularly run at less than 300 nanosecond access times making all but the FDDI real-time address routing practical with DRAM parts. Static RAM memories are currently available with 50 nanosecond and faster access times which makes even FDDI routing realizable. The address record memory 100 is only required to be 1/6 the speed of the index memory 98 since there are six octets between completion of the first and second record indexes. The source protect record 110 and the destination route record 112 feed into the buffered routing logic 56 of FIG. 2 to select the outbound path (MAC) for the message. ") Col. 17:40- Col. 18:8 ("The novel system uses arithmetic coding of the directory index 130 as shown by the diagrammatic illustration in FIG. 4. Arithmetic coding is a powerful technique for obtaining the near minimum entropy compression of a sequence of data bits. Since a network address is just a sequence of binary data bits of known length, the minimum entropy compression of all the combinations of bit strings represented by all of the active network addresses should produce the shortest number of bits which would uniquely identify all of the addresses. This encoding could then be used as an index 136 into the routing directory 130. Essentially arithmetic coding uses the distribution statistics of the symbols (in this case octet values) to divide a unit space into a unique fraction based on the sequence of symbols (octets) presented. As each symbol (octet) is presented, the unit space is subdivided into a smaller range. Symbols (Octets) with higher probability of occurrence reduce the range less than those with small probability, causing fewer bits to be used in encoding the higher probability octets….Thus in FIG. 4, the destination address 126 is compressed by the arithmetic code process 138 to obtain an integer 140 which represents the address. If further compression is needed, the integer can be compressed through truncation 142 by methods well-known in the art and further compressed if needed by

*Claim language of the ‘224 Patent (disputed terms in **bold**)*	*Plaintiff’s proposed construction and supporting evidence*	*Defendants’ proposed construction and supporting evidence*
		hashing 144, a technique also well-known in the art. The resulting index 136 is then used to find the unique address in the compressed address directory 130. The routing switch designs 38 and 96 shown in FIGS. 2 and 3 are specific implementations of the novel arithmetic compression process employed by this invention.”) *‘906 specification*: Figures 2-4 Abstract (“To provide fast access times with very large key fields, an associative memory utilizes a location addressable memory and lookup table to generate from a key the address in memory storing an associated record. The lookup tables, stored in memory, are constructed with the aid of arithmetic data compression methods to create a near perfect hashing of the keys.”) Col. 1:19-21 (“The present invention relates to associative memory systems, and more particularly to associative memory systems for handling large key set and spaced.”) Col. 4:51-56 (“Further, the present invention employs a reversible arithmetic code compression technique to reduce the logical network address of up to 128 bits to a unique integer value which preserves any hierarchical ordering of the network address.”) Col. 5:1-4 (“Arithmetic coding, when applied to addresses as known length keys, provides several advantages for table look-up when the addresses are known or can be learned in advance as they are in communications applications.”) Col. 5:18-35 (“Secondly, arithmetic coding can be constructed to operate on each symbol position in the address field as it arrives, allowing processing to begin as soon as the first address symbol arrives. Thirdly, arithmetic coding preserves the

*Claim language of the ‘224 Patent (disputed terms in **bold**)*	*Plaintiff’s proposed construction and supporting evidence*	*Defendants’ proposed construction and supporting evidence*
		hierarchical (left to right precedence) of the IS0 addresses being encoded. This is desirable if an Internet router only has knowledge of the network address but the Internet header carries the full destination address of a succeeding system node. Finally, a constant known set of computations is required for each symbol of the address field independent of the number of address symbols or the number of active Internet addresses. These features make the arithmetic coding used herein an ideal candidate for the routing table directory structure that is independent of a location address in a router, gate way or end-system.”) Col. 5:40-41 (“Addresses are just unique identification numbers represented by a string of symbols of known length. Each Internet router learns the location of these numbers within the network from the Internet protocol traffic, from the source addresses of the packets it receives, and from a network management protocol.”) Col. 6:4-7 (“The present invention combines arithmetic coding with dynamic hashing to provide a very high speed method and system for detecting the 48 bit physical addresses in a Media Access Controller (MAC). “) Col 6: 44-63 (“Another aspect of the invention is an apparatus and method for implementing a routing table directory to provide for fast access times to look up routing information. This apparatus is an application of a novel associative memory utilizing arithmetic coding to associate a key presented to the memory with a record stored in the memory, but has a very-wide range of application in many different types of data processing systems. The associative memory includes an index table stored in memory and a record memory for storing the records of data. The index table is constructed such that each symbol of a key, a key being divided into a string of symbols and each symbol being defined by its position within the key and its value, addresses an index value in the index table

*Claim language of the '224 Patent (disputed terms in **bold**)*	*Plaintiff's proposed construction and supporting evidence*	*Defendants' proposed construction and supporting evidence*
		memory. These index values are assigned such that the sum of index values for a given key is a unique value that is used to address the record memory. Several methods and apparatus are disclosed the permit random assignment of index values to new keys as they are presented, as well as for keys that are presented in sorted order for addition to the memory.") Col. 11:56-66 "To provide an Internet routing table that uses a flat logical address structure to provide fast and efficient route processing of both multicast and unicast message traffic. In the present system, the physical address structure is removed from the design and operation of the Internet routing by treating the message addresses as a symbol string without predetermined internal structure and processing them as if they are a unique identification code representing the host. This approach is made possible by employing an arithmetic code compression technique as a hashing function for the routing table access method." Col. 14:31-52 ("The address detection logic examines both the destination and source address fields represented by the octets shifted into buffer 48 and buffer 50. Six octets are in each buffer. When the twelve octets are all stored, each octet is used as an address into a 256 element index table for that address octet position. This requires six destination index tables 66 and six source index tables 68. The output of these tables (the contents of the location addressed by each octet) is then arithmetically combined in combiners 70 and 72. One method of arithmetically combining these outputs adds the six outputs of the source index table 68 to compute the source index 74. It also adds the six destination table 66 outputs to compute the route index 76. The source index 74 is used as the address into the source protect table 78 and the output of that location is the source protection record 80 which is coupled to the routing logic 82. Similarly, the route index 76 is used as the address of a location in the destination

*Claim language of the '224 Patent (disputed terms in **bold**)*	*Plaintiff's proposed construction and supporting evidence*	*Defendants' proposed construction and supporting evidence*
		routing table 84 and the contents of that location is coupled to route record 86. The outputs of the protect record 80 and route record 86 are used by the routing logic 56 in a well-known manner to determine which destination MAC is to receive the message.") Col. 15:12-22 ("The source protect table 78 in each node stores the source protect record 80 (also called the MultiCast Record List 134 in FIG, 4), which has information defining a shortest path from a particular source to that node, This shortest path information is computed from the messages received from forwarding nodes using a shortest path spanning tree algorithm well in the art. The source protect record 80 may be modified by management decision to prevent messages from a particular source identification code from being forwarded on particular paths to other nodes.") Col. 15:43-52 ("In FIG. 2, the address detection logic employs separate tables and arithmetic processing elements for both the source and destination address detection. While this approach allows the arithmetic processing and record table access to be relatively slow the slower elements are not sufficiently economical in price to be cost effective. Neither does the circuit of FIG. 2 utilize the fact that because the data octets arrive sequentially, they could be processed through the index look-up table and partial arithmetic computed each octet time. ") Col.15:53-col. 16:30 ("FIG. 3 is a circuit diagram of an alternate logic layout for serial processing of the incoming data by a switch 96 which is similar to switch 34. The data octets arrive sequentially and FIG. 3 discloses a logic layout which uses one bank of index table memory 98, one bank of address record memory 100 and one arithmetic computation unit 102 to accomplish both source and destination address detection. In this approach, the octet data bits are coupled serially into octet register 104 and are used as the low

*Claim language of the ‘224 Patent (disputed terms in **bold**)*	*Plaintiff’s proposed construction and supporting evidence*	*Defendants’ proposed construction and supporting evidence*
		order address bits into the index table 98. A byte counter 106 which counts the address octets from one to twelve as they arrive in the octet register 104 is used as the high order address bits into the index table 98. From byte count one to six, the arithmetic unit 102 partially computes the final index with each output from the index table. After byte count six, the computation for the destination address mask index is complete and transferred to the index buffer 108. The arithmetic unit 102 is then reset and the six octets of source address are computed. By the time the source protection record index 110 has been computed, the data in destination route record 112 has been loaded into its output buffer on line 114. The source protect record 110 is then accessed from a second bank of the mask memory 100 using the count twelve signal on line 116 as the high order address bit. This sequential detection approach shown in FIG. 3 places special performance requirements on the index table memory and each reiteration of the arithmetic computation. That is, the access to the index table 98 and the partial computation with the table output each must be complete in less than one octet time. However, the computation is delayed one octet clock time behind the table access. Each of these timing requirements is within the available speeds of commercially available VLSI computer memory and arithmetic components. Current DRAM memories regularly run at less than 300 nanosecond access times making all but the FDDI real-time address routing practical with DRAM parts. Static RAM memories are currently available with 50 nanosecond and faster access times which makes even FDDI routing realizable. The address record memory 100 is only required to be 1/6 the speed of the index memory 98 since there are six octets between completion of the first and second record indexes. The source protect record 110 and the destination route record 112 feed into the buffered routing logic 56 of FIG. 2 to select the outbound path (MAC) for the message. ”)

*Claim language of the '224 Patent (disputed terms in **bold**)*	*Plaintiff's proposed construction and supporting evidence*	*Defendants' proposed construction and supporting evidence*
		Col. 18:30-67 ("The novel system uses arithmetic coding of the directory index 130 as shown by the diagrammatic illustration in FIG. 4. Arithmetic coding is a powerful technique for obtaining the near minimum entropy compression of a sequence of data bits. Since a network address is just a sequence of binary data bits of known length, the minimum entropy compression of all the combinations of bit strings represented by all of the active network addresses should produce the shortest number of bits which would uniquely identify all of the addresses. This encoding could then be used as an index 136 into the routing directory 130. Essentially arithmetic coding uses the distribution statistics of the symbols (in this case octet values) to divide a unit space into a unique fraction based on the sequence of symbols (octets) presented. As each symbol (octet) is presented, the unit space is subdivided into a smaller range. Symbols (Octets) with higher probability of occurrence reduce the range less than those with small probability, causing fewer bits to be used in encoding the higher probability octets….Thus in FIG. 4, the destination address 126 is compressed by the arithmetic code process 138 to obtain an integer 140 which represents the address. If further compression is needed, the integer can be compressed through truncation 142 by methods well-known in the art and further compressed if needed by hashing 144, a technique also well-known in the art. The resulting index 136 is then used to find the unique address in the compressed address directory 130. The routing switch designs 38 and 96 shown in FIGS. 2 and 3 are specific implementations of the novel arithmetic compression process employed by this invention.") Col. 19:61-col. 21: 19 ("The address index tables (66, 68 or 98) are then incrementally filled in with sub-index values as particular address bit strings are encoded into the address tables. This processing takes the following steps. Initially the table (66, 68, and 198 in FIGS. 2 and 3) is entirely filled with zero entries

*Claim language of the '224 Patent (disputed terms in **bold**)*	*Plaintiff's proposed construction and supporting evidence*	*Defendants' proposed construction and supporting evidence*
		and the value of all locations in the table is set to zero. (1) A counter (Current_count) is established by the learned address logic 88 for each symbol position to keep track of the number of non-zero entries in this bank of the address index table and these counters are initially set to zero. In order to keep track of the number of addresses using a particular non-zero location in the address index table, a use counter is established in the learned address logic 88 for each non-zero location in each bank of the address index table. (2) The allowed maximum non-zero entries value for each symbol position is obtained from a management decision. (3) A range value (Range) is computed for each symbol position. The first range value is computed by setting the range for some symbol position to the allowed maximum count for that symbol position. The range value for the next symbol position is the range value for the previous symbol position times the allowed maximum count for this symbol position. The range value for each symbol position is the product of the range value of the next previously computed symbol position and the allowed maximum count for this symbol position. The order of the symbol positions used to compute the range values is only important in that the decoding operation used to recover the original address before encoding to an integer value must use the same symbol order as that used to compute the range values. The sequence of range value computations from the last address symbol to the first address symbol must be used to preserve hierarchical structure of the structure of the original address being encoded. Range (I)=Range (I+1) times Allowed_max_count (I) Each symbol from an address bit string to be encoded into the address index table is processed in the same sequence as that used to process the address symbols during receipt of the packets from a transmitter for routing table access. (1) Use the numeric value of the symbol as the address of the location in this symbol's bank of the address index table (66, 68 and

*Claim language of the ‘224 Patent (disputed terms in **bold**)*	*Plaintiff’s proposed construction and supporting evidence*	*Defendants’ proposed construction and supporting evidence*
		98). (a) If the existing entry in this location of this bank of the address index table is not zero, then increment the use count for this location and no further processing of this symbol is required and the next symbol may begin processing. (b) If the existing entry at this location in this bank of address index table is zero, then non-zero entry value is computed by (1) incrementing the current count for this symbol position, (2) checking to be sure the incremented current count is less than or equal to the allowed maximum count for this symbol position, and (3) (if the count is not greater than the maximum) computing the value of the incremented current count multiplied by the range value for this octet position and divided by the allowed maximum count for this position and storing this value in this location in the address index table and setting the use count for this location to “one”. If the incremented current count is greater than the allowed maximum count for this symbol position, then this address cannot be encoded into the address-index table and the management entity is notified that the address index table has overflowed unless another address is removed from the table making a use count go to “zero” and reducing the current count for this symbol position. (2) Continue processing address bit string symbols until the entire address has been encoded into the address index table by having for each symbol in the address a non-zero value for that symbol value location in every symbol position bank of the address index table. Address bit strings embedded in the incoming packets are compressed in the combine table outputs 70 and 72 in FIG. 2 and in arithmetic computation 102 of FIG. 3 to an integer value by adding together the stored values from the address index table bank for each symbol position where the symbol value is used as a location address into the bank for that symbol in the address index table. If any index table value accessed is zero, the processing stops and the zero detect 90 is activated. This zero indicates the address has not been encoded

*Claim language of the '224 Patent (disputed terms in **bold**)*	*Plaintiff's proposed construction and supporting evidence*	*Defendants' proposed construction and supporting evidence*
		into the address index table. If the number of significant bits in the encoded integer are larger than the size of the compressed address directory 130, then truncation 142 (removing some low order bits) and Modulo N hashing 144 (removing some of the high order bits) may be used to reduce the size of the encoded address integer to the number of locations in the compressed address directory 130.") Col. 24:34-45 ("When enabled, the arithmetic computation logic circuitry adds an index value on line 505 to a previously computed sum, in effect keeping a running total. When the running total exceeds P, P is subtracted from the running total. The arithmetic computation logic circuitry thus performs a Modulo (P) addition of the index values stored in the index table memory for each symbol in the key to create a final sum called a record index. The record index is a data value that will be used as a logical address to the place within key record memory 78 in which the record corresponding to the key presented on input lines 501 is stored.") *'224 patent prosecution history*: Applicant's First Amendment, page 5 ("New Claims 17-19 more clearly define the applicant's use of the arithmetic coding to generate a numerical address from a key of data pointing to a record storage location. Specifically, the applicant's invention utilizes a use count table to track the occurrence of each symbol within each symbol position. The use count table is then used to assign an index value to each symbol within each symbol position based upon the occurrence recorded within the use count table. Next, a key is presented to the applicant's invention the index values for the symbols within the key are determined and summed to create a numerical address point to the associated record location in record memory….[discusses the Leone reference coding]…the arithmetic coding/compression utilized by the applicants invention is distinctly different

Claim language of the '224 Patent (disputed terms in ***bold****)*	*Plaintiff's proposed construction and supporting evidence*	*Defendants' proposed construction and supporting evidence*
		from the coding scheme disclosed in Leone.") "Claim 33 [issued claim 1] has been amended to include the limitation formerly in claim 34 that looking up source filtering information includes arithmetically coding a source address to generate a unique identifier for a record containing source filtering information associated with the source address." p. 7 [Present patent claims added in April 21, 1997 amendment. (Correspondence between issued claim numbers and claim numbers used during prosecution: 1-33, 2-35, 3-46, 4-47, 6-48, 7-50, 8-51, 9-52, 10-53, 11-54, 12-55.)] Preliminary Amendment after CPA, paper 13, page 10 ("The claims pending in the application recite or are otherwise limited to processes and apparatus that 'arithmetically compress' or 'arithmetically code' a key to as a numerical number that points to a memory location holding a related record. 'Arithmetic coding' or 'arithmetic compression' is a generic term for a specific and well-defined coding scheme. It is defined in the specification and in several references that have been submitted… the applicant uses arithmetically coding to generate an index for an associative memory by arithmetically encoding keys as a numerical address to record associated with the key stored in a table.") Page 10 ("The claims pending in the application recite or are otherwise limited to processes and apparatus that 'arithmetically compress' or 'arithmetically code' a key to as a numerical number that points to a memory location holding a related record. 'Arithmetic coding' or 'arithmetic compression' is a generic term for a specific and well-defined coding scheme. It is defined in the specification and in several references that have been submitted… the applicant uses arithmetically coding to generate an index for an associative memory by

*Claim language of the ‘224 Patent (disputed terms in **bold**)*	*Plaintiff’s proposed construction and supporting evidence*	*Defendants’ proposed construction and supporting evidence*
		arithmetically encoding keys as a numerical address to record associated with the key stored in a table.”) Page 11 (“Each claim is limited to arithmetic coding or compression of keys.”) Appeal Brief, Paper 25, specifically: Page 3 (“[Summary of Invention] to provide for faster look-up of source filtering information, the logical source address is compressed, by arithmetically coding the source address, to an integer that uniquely identifies the record of data in memory containing the source filtering information.”) Page 5 (“The specification defines arithmetic coding.”) Page 5 (“Applicant utilizes arithmetically coding to generate an index to a record stored in memory by arithmetically encoding keys as a numerical address to a record associated with the key stored in a table.”) Pages 6-7 (Fenner argued with respect to all independent claims that “the use of arithmetic coding in looking up source filtering information is not shown in or suggested by Leone, et al. [the basis for a 102 rejection]”) Pages 6-7 (“In a telephonic interview with the Examiner on February 27, 1998, the Applicant's Attorney explained the foregoing to the Examiner, who expressed general agreement that arithmetic coding is a term of art that is distinguishable from other forms of hashing, particularly the CRC method, and indicated that he would now likely allow Claim 33, and the claims dependent therefrom, upon Applicant's submission of the foregoing arguments into the record.”) Page 10 (Applicant’s arithmetic coding is advantageous in that it is able, if desired, to

*Claim language of the ‘224 Patent (disputed terms in **bold**)*	*Plaintiff’s proposed construction and supporting evidence*	*Defendants’ proposed construction and supporting evidence*
		add keys without, in effect, having to find a define a new hash function each time or in the embodiment claim in claim 35, rebuilding from scratch an index table.”) July 8, 1994 Amendment, prosecution history of the '258 patent (To overcome a rejection based on the Leone reference, Fenner argued that arithmetic coding/compression is a specific technique and that the term as used in the disclosure for that patent does not encompass any kind of compression or truncation using an arithmetic operation.) *Prosecution History of Patent Application Serial No. 08/174361 (Related Application)*: Applicant’s Brief in Support of Appeal, paper #9, specifically, Pages 2-3 (“The invention pertains, generally, to associative memories in which a key is arithmetically coded as an integer that indentified a record of data in memory with which the key is associated…. Applicant utilizes arithmetically coding to generate an index to record stored in memory by arithmetically encoding keys as a numerical address to a record associated with the key stored in a table.”)
a physical media address for identifying a physical device for routing the data packet in physical media found in claim[s]: 8	a physical media address (as construed herein) for identifying a physical device for routing the data packet in physical media (as construed herein) <u>INTRINSIC EVIDENCE</u>: Column 10, lines 54-58; Column 1, lines 60-64; Column 7, lines 60-65; Column 12, lines 23-28	address in the physical destination address field <u>INTRINSIC EVIDENCE</u>: *‘224 specification*: Col. 5:65-6:1 (“The present invention combines arithmetic coding with dynamic hashing to provide a very high speed method and system for detecting the 48 bit physical addresses in a Media Access Controller (MAC).”). Col. 10:52-58 (“As stated, in the prior art, the aircraft 10 must have assigned to it a code representing its physical address with

*Claim language of the ‘224 Patent (disputed terms in **bold**)*	*Plaintiff’s proposed construction and supporting evidence*	*Defendants’ proposed construction and supporting evidence*
		respect to communication system 1. Physical addresses are associated with interface hardware. Thus, moving the hardware interface to a new machine or replacing a hardware interface that has failed changes the physical address of a particular host.”). Col. 11:31-39 (“This system then enables each node to store data representing the address of the last node communicating with a particular mobile vehicle and not the physical address of the vehicle. This allows communication from aircraft 10 to ship 12 throughout the various communication systems without either aircraft 10 or ship 12 being required to change network addresses as they move from access point to access point and without knowing the specific network location of the other.”). Col. 13:17-25 (“The Media Access Controller switch 38 transmits or forwards data it receives, and accepts data for transmission as eight parallel data bits called a data octet. It processes address symbols which are a fixed number of consecutive bits from the address bit string and may be from two to any number of bits in length. One size symbol is the “eight” bit octet which is the symbol size used in the address routing table circuits presented in FIGS. 2 and 3. The number of symbols in the maximum address length to be processed for a particular implementation is a design and management decision. The examples presented in FIGS. 2 and 3 use six octets as the maximum address length, since this is the length of the IEEE standard physical layer (MAC level) address used by Ethernet, Token Ring, and FDDI. The International Standards Organization (ISO) network layer (IP.ISO 124) employs a variable length, up to 20 octets, for the source and destination address 128 and *U6*, as shown in FIG. 4.”). Col. 16:22-31 (“The router strips off the incoming physical header 120 and the LLC header 122. It looks up the destination and source addresses 126 and 128 in its routing table 130, selects the appropriate outbound

*Claim language of the '224 Patent (disputed terms in **bold**)*	*Plaintiff's proposed construction and supporting evidence*	*Defendants' proposed construction and supporting evidence*
		link, or a plurality of outbound links in the case of a multicast group destination address from table 132 and reduces the plurality of outbound links using restrictions from table 134 in the case of a multicast transmission and passes the IP datagram packet to those selected channels for LLC encapsulation and transmission."). *'224 Prosecution History*: March 2, 1998 Brief in support of Appeal ("As discussed beginning on page 44, line 15, and continuing to page 46, line 2 of the application, there is a physical or media address in which a data packet is encapsulated, and a logical address for the source in the data packet which is independent of the source's physical or media address. During routing at each node that interconnects networks, the physical address layer is stripped off and replaced with the physical address of the node to which it is next forwarded."). March 2, 1998 Brief in support of Appeal ("Thus, source filtering may take place in the Internet environment, in which physical addresses of the source are stripped from a packet by an intermediary node, or with a mobile source which may access the data networks through different network media, and thus may not be assigned the same physical address during each access."). March 2, 1998 Brief in support of Appeal ("However, Leone, et al. does not mention or suggest filtering based on logical addresses of the source of the data packet. One benefit of filtering based on the logically address of the source is that it is device independent. It does not matter what network or device the source then currently resides or has accessed. Thus, source filtering may take place in the Internet environment, in which physical addresses of the source are stripped from a packet by an intermediary node, or with a mobile source which may access the data networks through different network media, and thus may not be assigned the same

*Claim language of the '224 Patent (disputed terms in **bold**)*	*Plaintiff's proposed construction and supporting evidence*	*Defendants' proposed construction and supporting evidence*
		physical address during each access.").
physical media address found in claim[s]: 8	address associated with the hardware of the physical media <u>INTRINSIC EVIDENCE</u>: Column 10, lines 54-58	address in the physical destination address field <u>INTRINSIC EVIDENCE</u>: Please refer to the intrinsic evidence reference above for the claim term "a physical media address for identifying a physical device for routing the data packet in physical media."
physical media found in claim[s]: 8	communication layer which controls the underlying hardware technologies <u>INTRINSIC EVIDENCE</u>: Column 1, lines 60-64; Column 7, lines 60-65; Column 12, lines 23-28	physical interconnection allowing the transfer of packets to and from the node <u>INTRINSIC EVIDENCE</u>: Please refer to the intrinsic evidence reference above for the claim term "a physical media address for identifying a physical device for routing the data packet in physical media."

2. U.S. Pat. No. 7,145,906

*Claim language of the '906 Patent (disputed terms in **bold**)*	*Plaintiff's proposed construction and supporting evidence*	*Defendants' proposed construction and supporting evidence*
MAC address found in claim[s]: 9, 10, 19, 20	physical address used by the media access controller (MAC) level defined by standards such as Ethernet, token ring, or FDDI <u>INTRINSIC EVIDENCE</u>: Column 14, lines 15-22; Column 6, lines 4-7	fixed, unique, and unchanging identifier assigned to a host <u>INTRINSIC EVIDENCE</u>: *'906 specification*: Col. 2:45-53 ("The present system overcomes the disadvantages of the prior art by simply assigning a fixed, unique and unchanging identification code to both host A and host B. As host B enters into a new network access system, it transmits its identification code to the nearest node and all of the nodes interconnecting all of the disparate networks each store, with the unique identification code of host B, the address of those nodes which can communicate with host B so that a path can be completed through the nodes between

*Claim language of the '906 Patent (disputed terms in **bold**)*	*Plaintiff's proposed construction and supporting evidence*	*Defendants' proposed construction and supporting evidence*
		host A and host B.") Col. 9:20-24 ("The present invention modifies the system of FIG. 1 to overcome the disadvantages of the prior art by allowing each host to have a fixed unique identification code instead of an address code which changes to identify itself with whatever communication network it may operating [sic]."). Col. 11:55-12:5 ("The novel system of the present invention modifies FIG. 1 to provide an Internet routing table that uses a flat logical address structure to provide fast and efficient route processing of both multicast and unicast message traffic. In the present system, the physical address structure is removed from the design and operation of the Internet routing by treating the message addresses as a symbol string without predetermined internal structure and processing them as if they are a unique identification code representing the host. This approach is made possible by employing an arithmetic code compression technique as a hashing function for the routing table access method. By managing and manipulating logical network addresses within the system, mobile end systems can keep the same network identification code (not physical address) as they move from communication network to communication network. Similarly, group or multicast addresses may be allocated without regard to their physical network connection."). Col. 5:36-45 ("The present invention provides a very fast, automatically expandable, source filtered Internet routing scheme totally independent of the internal logical or physical structure of the network addresses in the message format that it is routing. Addresses are just unique identification numbers represented by a string of symbols of known length. Each Internet router learns the location of these numbers within the network from the Internet protocol traffic, from the source addresses of the packets it receives, and from a network management protocol.").

*Claim language of the '906 Patent (disputed terms in **bold**)*	*Plaintiff's proposed construction and supporting evidence*	*Defendants' proposed construction and supporting evidence*
		Col. 9:41-48 ("If aircraft 10 desires to contact ship 12, it simply transmits a message format including its own unique code and the unique identification code for ship 12 to the nearest system 14. The receiving system 14 sends the message to node 18 which checks its memory tables to determine if it has stored the address of the last node (26 or 32) communicating with ship 12. If not, it stores the unique identification code of aircraft 10."). Col. 11:22-26 ("The problem with such vehicle movement with the prior art system, as stated, is that each of the communication systems 1-5 are different networks and may use different types of media access protocols for operation which require the network address of the moving vehicle to be changed."). Col. 11:39-45 ("As stated, in the prior art, the aircraft 10 must have assigned to it a code representing its physical address with respect to communication system 1. Physical addresses are associated with interface hardware. Thus, moving the hardware interface to a new machine or replacing a hardware interface that has failed changes the physical address of a particular host."). Fig. 2; col. 13:50-15:51 (Fig. 2 and related descriptions). Col. 9:5-19 ("However, when a host passes from one communication system to another, the address code of that host must be changed to be conformed with or admitted to the new communication system. Thus, if a host passes from an FDDI to an ETHERNET system, the address code of the host must be changed in order to enable the new system to accommodate it. This change may require a great deal of manipulation of data within the system and require expensive additional equipment to enable the appropriate changes to be made. Further, by the time one host (ship 12) sends a message to the last known address of the

*Claim language of the '906 Patent (disputed terms in **bold**)*	*Plaintiff's proposed construction and supporting evidence*	*Defendants' proposed construction and supporting evidence*
		moving host (aircraft 10), the 15 moving host may have entered the range of a new communication network and have a different address code thereby causing a problem in receiving the message sent with a network dependent address from the message sending host.").
each communications port having associated with it a MAC address found in claim[s]: 9, 10, 19, 20	each communications port having associated with it a MAC address (as construed herein) INTRINSIC EVIDENCE: Column 14, lines 15-22; Column 6, lines 4-7	each communications port is referenced in a record identified by a unique value created by arithmetically compressing, as distinct from hashing, a MAC address INTRINSIC EVIDENCE: *'906 specification*: Figures 2-4 Abstract ("To provide fast access times with very large key fields, an associative memory utilizes a location addressable memory and lookup table to generate from a key the address in memory storing an associated record. The lookup tables, stored in memory, are constructed with the aid of arithmetic data compression methods to create a near perfect hashing of the keys.") Col. 1:19-21 ("The present invention relates to associative memory systems, and more particularly to associative memory systems for handling large key set and spaced.") Col. 4:51-56 ("Further, the present invention employs a reversible arithmetic code compression technique to reduce the logical network address of up to 128 bits to a unique integer value which preserves any hierarchical ordering of the network address.") Col. 5:1-4 ("Arithmetic coding, when applied to addresses as known length keys, provides several advantages for table look-up when the addresses are known or can be learned in advance as they are in communications applications.") Col. 5:18-35 ("Secondly, arithmetic coding

*Claim language of the '906 Patent (disputed terms in **bold**)*	*Plaintiff's proposed construction and supporting evidence*	*Defendants' proposed construction and supporting evidence*
		can be constructed to operate on each symbol position in the address field as it arrives, allowing processing to begin as soon as the first address symbol arrives. Thirdly, arithmetic coding preserves the hierarchical (left to right precedence) of the IS0 addresses being encoded. This is desirable if an Internet router only has knowledge of the network address but the Internet header carries the full destination address of a succeeding system node. Finally, a constant known set of computations is required for each symbol of the address field independent of the number of address symbols or the number of active Internet addresses. These features make the arithmetic coding used herein an ideal candidate for the routing table directory structure that is independent of a location address in a router, gate way or end-system.") Col. 5:40-41 ("Addresses are just unique identification numbers represented by a string of symbols of known length. Each Internet router learns the location of these numbers within the network from the Internet protocol traffic, from the source addresses of the packets it receives, and from a network management protocol.") Col. 6:4-7 ("The present invention combines arithmetic coding with dynamic hashing to provide a very high speed method and system for detecting the 48 bit physical addresses in a Media Access Controller (MAC). ") Col 6: 44-63 ("Another aspect of the invention is an apparatus and method for implementing a routing table directory to provide for fast access times to look up routing information. This apparatus is an application of a novel associative memory utilizing arithmetic coding to associate a key presented to the memory with a record stored in the memory, but has a very-wide range of application in many different types of data processing systems. The associative memory includes an index table stored in memory and a record memory for storing the records of data. The index table

*Claim language of the '906 Patent (disputed terms in **bold**)*	*Plaintiff's proposed construction and supporting evidence*	*Defendants' proposed construction and supporting evidence*
		is constructed such that each symbol of a key, a key being divided into a string of symbols and each symbol being defined by its position within the key and its value, addresses an index value in the index table memory. These index values are assigned such that the sum of index values for a given key is a unique value that is used to address the record memory. Several methods and apparatus are disclosed the permit random assignment of index values to new keys as they are presented, as well as for keys that are presented in sorted order for addition to the memory.") Col. 11:56-66 "To provide an Internet routing table that uses a flat logical address structure to provide fast and efficient route processing of both multicast and unicast message traffic. In the present system, the physical address structure is removed from the design and operation of the Internet routing by treating the message addresses as a symbol string without predetermined internal structure and processing them as if they are a unique identification code representing the host. This approach is made possible by employing an arithmetic code compression technique as a hashing function for the routing table access method." Col. 14:31-52 ("The address detection logic examines both the destination and source address fields represented by the octets shifted into buffer 48 and buffer 50. Six octets are in each buffer. When the twelve octets are all stored, each octet is used as an address into a 256 element index table for that address octet position. This requires six destination index tables 66 and six source index tables 68. The output of these tables (the contents of the location addressed by each octet) is then arithmetically combined in combiners 70 and 72. One method of arithmetically combining these outputs adds the six outputs of the source index table 68 to compute the source index 74. It also adds the six destination table 66 outputs to compute the route index 76. The source index 74 is used as the address into the

*Claim language of the '906 Patent (disputed terms in **bold**)*	*Plaintiff's proposed construction and supporting evidence*	*Defendants' proposed construction and supporting evidence*
		source protect table 78 and the output of that location is the source protection record 80 which is coupled to the routing logic 82. Similarly, the route index 76 is used as the address of a location in the destination routing table 84 and the contents of that location is coupled to route record 86. The outputs of the protect record 80 and route record 86 are used by the routing logic 56 in a well-known manner to determine which destination MAC is to receive the message.") Col. 15:12-22 ("The source protect table 78 in each node stores the source protect record 80 (also called the MultiCast Record List 134 in FIG, 4), which has information defining a shortest path from a particular source to that node, This shortest path information is computed from the messages received from forwarding nodes using a shortest path spanning tree algorithm well in the art. The source protect record 80 may be modified by management decision to prevent messages from a particular source identification code from being forwarded on particular paths to other nodes.") Col. 15:43-52 ("In FIG. 2, the address detection logic employs separate tables and arithmetic processing elements for both the source and destination address detection. While this approach allows the arithmetic processing and record table access to be relatively slow the slower elements are not sufficiently economical in price to be cost effective. Neither does the circuit of FIG. 2 utilize the fact that because the data octets arrive sequentially, they could be processed through the index look-up table and partial arithmetic computed each octet time. ") Col.15:53-col. 16:30 ("FIG. 3 is a circuit diagram of an alternate logic layout for serial processing of the incoming data by a switch 96 which is similar to switch 34. The data octets arrive sequentially and FIG. 3 discloses a logic layout which uses one bank of index table memory 98, one bank of address record memory 100 and

*Claim language of the '906 Patent (disputed terms in **bold**)*	*Plaintiff's proposed construction and supporting evidence*	*Defendants' proposed construction and supporting evidence*
		one arithmetic computation unit 102 to accomplish both source and destination address detection. In this approach, the octet data bits are coupled serially into octet register 104 and are used as the low order address bits into the index table 98. A byte counter 106 which counts the address octets from one to twelve as they arrive in the octet register 104 is used as the high order address bits into the index table 98. From byte count one to six, the arithmetic unit 102 partially computes the final index with each output from the index table. After byte count six, the computation for the destination address mask index is complete and transferred to the index buffer 108. The arithmetic unit 102 is then reset and the six octets of source address are computed. By the time the source protection record index 110 has been computed, the data in destination route record 112 has been loaded into its output buffer on line 114. The source protect record 110 is then accessed from a second bank of the mask memory 100 using the count twelve signal on line 116 as the high order address bit. This sequential detection approach shown in FIG. 3 places special performance requirements on the index table memory and each reiteration of the arithmetic computation. That is, the access to the index table 98 and the partial computation with the table output each must be complete in less than one octet time. However, the computation is delayed one octet clock time behind the table access. Each of these timing requirements is within the available speeds of commercially available VLSI computer memory and arithmetic components. Current DRAM memories regularly run at less than 300 nanosecond access times making all but the FDDI real-time address routing practical with DRAM parts. Static RAM memories are currently available with 50 nanosecond and faster access times which makes even FDDI routing realizable. The address record memory 100 is only required to be 1/6 the speed of the index memory 98 since there are six octets between completion of the first and second record indexes. The source protect record

*Claim language of the '906 Patent (disputed terms in **bold**)*	*Plaintiff's proposed construction and supporting evidence*	*Defendants' proposed construction and supporting evidence*
		110 and the destination route record 112 feed into the buffered routing logic 56 of FIG. 2 to select the outbound path (MAC) for the message. ") Col. 18:30-67 ("The novel system uses arithmetic coding of the directory index 130 as shown by the diagrammatic illustration in FIG. 4. Arithmetic coding is a powerful technique for obtaining the near minimum entropy compression of a sequence of data bits. Since a network address is just a sequence of binary data bits of known length, the minimum entropy compression of all the combinations of bit strings represented by all of the active network addresses should produce the shortest number of bits which would uniquely identify all of the addresses. This encoding could then be used as an index 136 into the routing directory 130. Essentially arithmetic coding uses the distribution statistics of the symbols (in this case octet values) to divide a unit space into a unique fraction based on the sequence of symbols (octets) presented. As each symbol (octet) is presented, the unit space is subdivided into a smaller range. Symbols (Octets) with higher probability of occurrence reduce the range less than those with small probability, causing fewer bits to be used in encoding the higher probability octets….Thus in FIG. 4, the destination address 126 is compressed by the arithmetic code process 138 to obtain an integer 140 which represents the address. If further compression is needed, the integer can be compressed through truncation 142 by methods well-known in the art and further compressed if needed by hashing 144, a technique also well-known in the art. The resulting index 136 is then used to find the unique address in the compressed address directory 130. The routing switch designs 38 and 96 shown in FIGS. 2 and 3 are specific implementations of the novel arithmetic compression process employed by this invention.") Col. 19:61-col. 21: 19 ("The address index tables (66, 68 or 98) are then incrementally filled in with sub-index values as particular

*Claim language of the ‘906 Patent (disputed terms in **bold**)*	*Plaintiff’s proposed construction and supporting evidence*	*Defendants’ proposed construction and supporting evidence*
		address bit strings are encoded into the address tables. This processing takes the following steps. Initially the table (66, 68, and 198 in FIGS. 2 and 3) is entirely filled with zero entries and the value of all locations in the table is set to zero. (1) A counter (Current_count) is established by the learned address logic 88 for each symbol position to keep track of the number of non-zero entries in this bank of the address index table and these counters are initially set to zero. In order to keep track of the number of addresses using a particular non-zero location in the address index table, a use counter is established in the learned address logic 88 for each non-zero location in each bank of the address index table. (2) The allowed maximum non-zero entries value for each symbol position is obtained from a management decision. (3) A range value (Range) is computed for each symbol position. The first range value is computed by setting the range for some symbol position to the allowed maximum count for that symbol position. The range value for the next symbol position is the range value for the previous symbol position times the allowed maximum count for this symbol position. The range value for each symbol position is the product of the range value of the next previously computed symbol position and the allowed maximum count for this symbol position. The order of the symbol positions used to compute the range values is only important in that the decoding operation used to recover the original address before encoding to an integer value must use the same symbol order as that used to compute the range values. The sequence of range value computations from the last address symbol to the first address symbol must be used to preserve hierarchical structure of the structure of the original address being encoded. Range (I)=Range (I+1) times Allowed_max_count (I) Each symbol from an address bit string to be encoded into the address index table is processed in the same sequence as that used to process the address symbols during

*Claim language of the '906 Patent (disputed terms in **bold**)*	*Plaintiff's proposed construction and supporting evidence*	*Defendants' proposed construction and supporting evidence*
		receipt of the packets from a transmitter for routing table access. (1) Use the numeric value of the symbol as the address of the location in this symbol's bank of the address index table (66, 68 and 98). (a) If the existing entry in this location of this bank of the address index table is not zero, then increment the use count for this location and no further processing of this symbol is required and the next symbol may begin processing. (b) If the existing entry at this location in this bank of address index table is zero, then non-zero entry value is computed by (1) incrementing the current count for this symbol position, (2) checking to be sure the incremented current count is less than or equal to the allowed maximum count for this symbol position, and (3) (if the count is not greater than the maximum) computing the value of the incremented current count multiplied by the range value for this octet position and divided by the allowed maximum count for this position and storing this value in this location in the address index table and setting the use count for this location to "one". If the incremented current count is greater than the allowed maximum count for this symbol position, then this address cannot be encoded into the address-index table and the management entity is notified that the address index table has overflowed unless another address is removed from the table making a use count go to "zero" and reducing the current count for this symbol position. (2) Continue processing address bit string symbols until the entire address has been encoded into the address index table by having for each symbol in the address a non-zero value for that symbol value location in every symbol position bank of the address index table. Address bit strings embedded in the incoming packets are compressed in the combine table outputs 70 and 72 in FIG. 2 and in arithmetic computation 102 of FIG. 3 to an integer value by adding together the stored values from the address index table bank for each symbol position where the symbol value is used as a location address

*Claim language of the '906 Patent (disputed terms in **bold**)*	*Plaintiff's proposed construction and supporting evidence*	*Defendants' proposed construction and supporting evidence*
		into the bank for that symbol in the address index table. If any index table value accessed is zero, the processing stops and the zero detect 90 is activated. This zero indicates the address has not been encoded into the address index table. If the number of significant bits in the encoded integer are larger than the size of the compressed address directory 130, then truncation 142 (removing some low order bits) and Modulo N hashing 144 (removing some of the high order bits) may be used to reduce the size of the encoded address integer to the number of locations in the compressed address directory 130.") Col. 24:34-45 ("When enabled, the arithmetic computation logic circuitry adds an index value on line 505 to a previously computed sum, in effect keeping a running total. When the running total exceeds P, P is subtracted from the running total. The arithmetic computation logic circuitry thus performs a Modulo (P) addition of the index values stored in the index table memory for each symbol in the key to create a final sum called a record index. The record index is a data value that will be used as a logical address to the place within key record memory 78 in which the record corresponding to the key presented on input lines 501 is stored.") *'224 specification*: Figures 2-4 Abstract ("To provide for fast access times with very large key fields, an associative memory utilizes a location addressable memory and look up tables to generate from a key an address in memory storing an associated record. The look up tables, stored in a memory, are constructed with the aid of arithmetic data compression methods to create a near perfect hashing of the keys. For encoding into the look up table, keys are divided into a string of symbols. Each symbol is assigned an index value, such that a modulo sum of index values for symbols of a particular key is a unique value that is used as an address to

*Claim language of the ‘906 Patent (disputed terms in **bold**)*	*Plaintiff’s proposed construction and supporting evidence*	*Defendants’ proposed construction and supporting evidence*
		the memory storing the record associated with that key.”) Col. 1:13-15 (“The present invention relates to associative memory systems, and more particularly to associative memory systems for handling large key set and spaced.”) Col. 4:45-49 (“Further, the present invention employs a reversible arithmetic code compression technique to reduce the logical network address of up to 128 bits to a unique integer value which preserves any hierarchical ordering of the network address.”) Col. 4:62-65 (“Arithmetic coding, when applied to addresses as known length keys, provides several advantages for table look-up when the addresses are known or can be learned in advance as they are in communications applications.”) Col. 5:12-29 (“Secondly, arithmetic coding can be constructed to operate on each symbol position in the address field as it arrives, allowing processing to begin as soon as the first address symbol arrives. Thirdly, arithmetic coding preserves the hierarchical (left to right precedence) of the IS0 addresses being encoded. This is desirable if an Internet router only has knowledge of the network address but the Internet header carries the full destination address of a succeeding system node. Finally, a constant known set of computations is required for each symbol of the address field independent of the number of address symbols or the number of active Internet addresses. These features make the arithmetic coding used herein an ideal candidate for the routing table directory structure that is independent of a location address in a router, gate way or end-system.”) Col. 5:34-39 (“Addresses are just unique

*Claim language of the '906 Patent (disputed terms in **bold**)*	*Plaintiff's proposed construction and supporting evidence*	*Defendants' proposed construction and supporting evidence*
		identification numbers represented by a string of symbols of known length. Each Internet router learns the location of these numbers within the network from the Internet protocol traffic, from the source addresses of the packets it receives, and from a network management protocol.") Col. 5:65- Col. 6:1 ("The present invention combines arithmetic coding with dynamic hashing to provide a very high speed method and system for detecting the 48 bit physical addresses in a Media Access Controller (MAC). ") Col 6: 37-56 ("Another aspect of the invention is an apparatus and method for implementing a routing table directory to provide for fast access times to look up routing information. This apparatus is an application of a novel associative memory utilizing arithmetic coding to associate a key presented to the memory with a record stored in the memory, but has a very-wide range of application in many different types of data processing systems. The associative memory includes an index table stored in memory and a record memory for storing the records of data. The index table is constructed such that each symbol of a key, a key being divided into a string of symbols and each symbol being defined by its position within the key and its value, addresses an index value in the index table memory. These index values are assigned such that the sum of index values for a given key is a unique value that is used to address the record memory. Several methods and apparatus are disclosed the permit random assignment of index values to new keys as they are presented, as well as for keys that are presented in sorted order for addition to the memory.") Col. 11:1-12 "[T]o provide an Internet routing table that uses a flat logical address structure to provide fast and efficient route processing of both multicast and unicast message traffic. In the present system, the physical address structure is removed from the design and operation of the Internet routing by treating the message addresses

*Claim language of the '906 Patent (disputed terms in **bold**)*	*Plaintiff's proposed construction and supporting evidence*	*Defendants' proposed construction and supporting evidence*
		as a symbol string without predetermined internal structure and processing them as if they are a unique identification code representing the host. This approach is made possible by employing an arithmetic code compression technique as a hashing function for the routing table access method." Col. 13:44-65 ("The address detection logic examines both the destination and source address fields represented by the octets shifted into buffer 48 and buffer 50. Six octets are in each buffer. When the twelve octets are all stored, each octet is used as an address into a 256 element index table for that address octet position. This requires six destination index tables 66 and six source index tables 68. The output of these tables (the contents of the location addressed by each octet) is then arithmetically combined in combiners 70 and 72. One method of arithmetically combining these outputs adds the six outputs of the source index table 68 to compute the source index 74. It also adds the six destination table 66 outputs to compute the route index 76. The source index 74 is used as the address into the source protect table 78 and the output of that location is the source protection record 80 which is coupled to the routing logic 82. Similarly, the route index 76 is used as the address of a location in the destination routing table 84 and the contents of that location is coupled to route record 86. The outputs of the protect record 80 and route record 86 are used by the routing logic 56 in a well-known manner to determine which destination MAC is to receive the message.") Col. 14:56-65 ("In FIG. 2, the address detection logic employs separate tables and arithmetic processing elements for both the source and destination address detection. While this approach allows the arithmetic processing and record table access to be relatively slow the slower elements are not sufficiently economical in price to be cost effective. Neither does the circuit of FIG. 2 utilize the fact that because the data octets

*Claim language of the ‘906 Patent (disputed terms in **bold**)*	*Plaintiff’s proposed construction and supporting evidence*	*Defendants’ proposed construction and supporting evidence*
		arrive sequentially, they could be processed through the index look-up table and partial arithmetic computed each octet time. “) Col. 14:66-col. 15:42 (“FIG. 3 is a circuit diagram of an alternate logic layout for serial processing of the incoming data by a switch 96 which is similar to switch 34. The data octets arrive sequentially and FIG. 3 discloses a logic layout which uses one bank of index table memory 98, one bank of address record memory 100 and one arithmetic computation unit 102 to accomplish both source and destination address detection. In this approach, the octet data bits are coupled serially into octet register 104 and are used as the low order address bits into the index table 98. A byte counter 106 which counts the address octets from one to twelve as they arrive in the octet register 104 is used as the high order address bits into the index table 98. From byte count one to six, the arithmetic unit 102 partially computes the final index with each output from the index table. After byte count six, the computation for the destination address mask index is complete and transferred to the index buffer 108. The arithmetic unit 102 is then reset and the six octets of source address are computed. By the time the source protection record index 110 has been computed, the data in destination route record 112 has been loaded into its output buffer on line 114. The source protect record 110 is then accessed from a second bank of the mask memory 100 using the count twelve signal on line 116 as the high order address bit. This sequential detection approach shown in FIG. 3 places special performance requirements on the index table memory and each reiteration of the arithmetic computation. That is, the access to the index table 98 and the partial computation with the table output each must be complete in less than one octet time. However, the computation is delayed one octet clock time behind the table access. Each of these timing requirements is within the available speeds of commercially available VLSI computer

*Claim language of the '906 Patent (disputed terms in **bold**)*	*Plaintiff's proposed construction and supporting evidence*	*Defendants' proposed construction and supporting evidence*
		memory and arithmetic components. Current DRAM memories regularly run at less than 300 nanosecond access times making all but the FDDI real-time address routing practical with DRAM parts. Static RAM memories are currently available with 50 nanosecond and faster access times which makes even FDDI routing realizable. The address record memory 100 is only required to be 1/6 the speed of the index memory 98 since there are six octets between completion of the first and second record indexes. The source protect record 110 and the destination route record 112 feed into the buffered routing logic 56 of FIG. 2 to select the outbound path (MAC) for the message. ") Col. 17:40- Col. 18:8 ("The novel system uses arithmetic coding of the directory index 130 as shown by the diagrammatic illustration in FIG. 4. Arithmetic coding is a powerful technique for obtaining the near minimum entropy compression of a sequence of data bits. Since a network address is just a sequence of binary data bits of known length, the minimum entropy compression of all the combinations of bit strings represented by all of the active network addresses should produce the shortest number of bits which would uniquely identify all of the addresses. This encoding could then be used as an index 136 into the routing directory 130. Essentially arithmetic coding uses the distribution statistics of the symbols (in this case octet values) to divide a unit space into a unique fraction based on the sequence of symbols (octets) presented. As each symbol (octet) is presented, the unit space is subdivided into a smaller range. Symbols (Octets) with higher probability of occurrence reduce the range less than those with small probability, causing fewer bits to be used in encoding the higher probability octets….Thus in FIG. 4, the destination address 126 is compressed by the arithmetic code process 138 to obtain an integer 140 which represents the address. If further compression is needed, the integer can be compressed through truncation 142 by methods well-known in

Claim language of the ‘906 Patent (disputed terms in ***bold****)*	*Plaintiff’s proposed construction and supporting evidence*	*Defendants’ proposed construction and supporting evidence*
		the art and further compressed if needed by hashing 144, a technique also well-known in the art. The resulting index 136 is then used to find the unique address in the compressed address directory 130. The routing switch designs 38 and 96 shown in FIGS. 2 and 3 are specific implementations of the novel arithmetic compression process employed by this invention.”) *‘224 patent prosecution history*: Applicant’s First Amendment, page 5 (“New Claims 17-19 more clearly define the applicant’s use of the arithmetic coding to generate a numerical address from a key of data pointing to a record storage location. Specifically, the applicant’s invention utilizes a use count table to track the occurrence of each symbol within each symbol position. The use count table is then used to assign an index value to each symbol within each symbol position based upon the occurrence recorded within the use count table. Next, a key is presented to the applicant’s invention the index values for the symbols within the key are determined and summed to create a numerical address point to the associated record location in record memory….[discusses the Leone reference coding]…the arithmetic coding/compression utilized by the applicants invention is distinctly different from the coding scheme disclosed in Leone.”) “Claim 33 [issued claim 1] has been amended to include the limitation formerly in claim 34 that looking up source filtering information includes arithmetically coding a source address to generate a unique identifier for a record containing source filtering information associated with the source address.” p. 7 [Present patent claims added in April 21, 1997 amendment. (Correspondence between issued claim numbers and claim numbers used during prosecution: 1-33, 2-35, 3-46, 4-47, 6-48, 7-50, 8-51, 9-52, 10-53, 11-54, 12-55.)] Preliminary Amendment after CPA, paper

*Claim language of the '906 Patent (disputed terms in **bold**)*	*Plaintiff's proposed construction and supporting evidence*	*Defendants' proposed construction and supporting evidence*
		13, page 10 ("The claims pending in the application recite or are otherwise limited to processes and apparatus that 'arithmetically compress' or 'arithmetically code' a key to as a numerical number that points to a memory location holding a related record. 'Arithmetic coding' or 'arithmetic compression' is a generic term for a specific and well-defined coding scheme. It is defined in the specification and in several references that have been submitted… the applicant uses arithmetically coding to generate an index for an associative memory by arithmetically encoding keys as a numerical address to record associated with the key stored in a table.") Page 10 ("The claims pending in the application recite or are otherwise limited to processes and apparatus that 'arithmetically compress' or 'arithmetically code' a key to as a numerical number that points to a memory location holding a related record. 'Arithmetic coding' or 'arithmetic compression' is a generic term for a specific and well-defined coding scheme. It is defined in the specification and in several references that have been submitted… the applicant uses arithmetically coding to generate an index for an associative memory by arithmetically encoding keys as a numerical address to record associated with the key stored in a table.") Page 11 ("Each claim is limited to arithmetic coding or compression of keys.") Appeal Brief, Paper 25, specifically: Page 3 ("[Summary of Invention] to provide for faster look-up of source filtering information, the logical source address is compressed, by arithmetically coding the source address, to an integer that uniquely identifies the record of data in memory containing the source filtering

*Claim language of the '906 Patent (disputed terms in **bold**)*	*Plaintiff's proposed construction and supporting evidence*	*Defendants' proposed construction and supporting evidence*
		information.") Page 5 ("The specification defines arithmetic coding.") Page 5 ("Applicant utilizes arithmetically coding to generate an index to a record stored in memory by arithmetically encoding keys as a numerical address to a record associated with the key stored in a table.") Pages 6-7 (Fenner argued with respect to all independent claims that "the use of arithmetic coding in looking up source filtering information is not shown in or suggested by Leone, et al. [the basis for a 102 rejection]") Pages 6-7 ("In a telephonic interview with the Examiner on February 27, 1998, the Applicant's Attorney explained the foregoing to the Examiner, who expressed general agreement that arithmetic coding is a term of art that is distinguishable from other forms of hashing, particularly the CRC method, and indicated that he would now likely allow Claim 33, and the claims dependent therefrom, upon Applicant's submission of the foregoing arguments into the record.") Page 10 (Applicant's arithmetic coding is advantageous in that it is able, if desired, to add keys without, in effect, having to find a define a new hash function each time or in the embodiment claim in claim 35, rebuilding from scratch an index table.") July 8, 1994 Amendment, prosecution history of the '258 patent (To overcome a rejection based on the Leone reference, Fenner argued that arithmetic coding/compression is a specific technique and that the term as used in the disclosure for that patent does not encompass any kind of compression or truncation using an arithmetic operation.) *Prosecution History of Patent Application Serial No. 08/174361 (Related*

*Claim language of the '906 Patent (disputed terms in **bold**)*	*Plaintiff's proposed construction and supporting evidence*	*Defendants' proposed construction and supporting evidence*
		Application): Applicant's Brief in Support of Appeal, paper #9, specifically, Pages 2-3 ("The invention pertains, generally, to associative memories in which a key is arithmetically coded as an integer that indentified a record of data in memory with which the key is associated…. Applicant utilizes arithmetically coding to generate an index to record stored in memory by arithmetically encoding keys as a numerical address to a record associated with the key stored in a table.")
stored association with one of the three least communications ports found in claim[s]: 9, 10, 19, 20	No construction needed	stored reference to the one of the at least three communications ports in a record identified by a unique value created by arithmetically compressing, as distinct from hashing, a MAC address <u>INTRINSIC EVIDENCE</u>: Please refer to the intrinsic evidence reference above for the claim term "each communications port having associated with it a MAC address."
if the source address filtering information is associated with the first MAC address found in claim[s]: 9, 19	if the source address filtering information (as construed herein) is associated with the first MAC address (as construed herein) <u>INTRINSIC EVIDENCE</u>: Column 13, lines 54-59; Column 14, lines 15-22; Column 6, lines 4-7	if source address filtering information (as construed herein) is in a record identified by a unique value created by arithmetically compressing, as distinct from hashing, the first MAC address <u>INTRINSIC EVIDENCE</u>: Please refer to the intrinsic evidence reference above for the claim term "each communications port having associated with it a MAC address."
associated with a stored protection record indicating protection of that communications port from packets containing the first MAC address as a	associated with a stored protection record (as construed herein) of that communications port from packets containing the first MAC address (as construed herein) as a MAC source address (as construed herein)	referenced in a stored protection record (as construed herein), identified by arithmetically compressing, as distinct from hashing, the first MAC address, indicating that the communications port is not allowed to forward packets containing the first MAC address as a MAC source

*Claim language of the '906 Patent (disputed terms in **bold**)*	*Plaintiff's proposed construction and supporting evidence*	*Defendants' proposed construction and supporting evidence*
MAC source address found in claim[s]: 10, 20	<u>INTRINSIC EVIDENCE</u>: Column 14, lines 15-22; Column 6, lines 4-7; Figure 4; Column 5, lines 41-45; Column 7, lines 31-38; Column 12, lines 55-57	address <u>INTRINSIC EVIDENCE</u>: Please refer to the intrinsic evidence reference above for the claim term "each communications port having associated with it a MAC address."

B. Claim Elements the Parties Agree Should be Governed by 35 U.S.C. § 112 ¶ 6, But in Dispute as to Which Structures, Acts, or Materials the Elements Correspond

1. U.S. Pat. No. 5,842,224

*Claim element (disputed terms in **bold**)*[1]	*Plaintiff's proposed structures, acts, or materials to which the elements correspond*	*Defendants' proposed structures, acts, or materials to which the elements correspond*
means for receiving a data packet [the data packet including a physical media address for identifying a physical device for routing the data packet in physical media and a source address for logically identifying a sender of the data packet independent of the sender's physical media address] found in claim[s]: 8	The means for receiving a data packet are at least the Media Access Controllers (Figure 2 – items 34, 40, 42, 44, 46). See construction for "means for receiving a data packet." Defendants have identified this phrase pursuant to 35 U.S.C. § 112 ¶ 6. The remainder of the claim limitation following "means for receiving a data packet" merely describes the "data packet" and not the structure for receiving a data packet.	FIG 1: Receivers/Receiving Systems/Radio Receivers 14, 18, 22, and 28. Defendants contend that the entire element, including the bracketed material, is governed by 35 U.S.C. § 112 ¶ 6, such that only those structures, acts or materials capable of receiving the data packet described in the claim correspond to this element.
means for looking up in a directory table stored at the controller using the source address source filtering information associated with the source address	The means for looking up in a directory table are at least the source index, the source protect table and the protect record (Figure 2 – items 74, 78, 80). The claim language itself describes how source filtering information (as construed herein) associated with the source address (as construed herein) is used for looking up in	FIG 2: Source Address Index Table 68; Source Index Table 74; Source Protect Table 78; Source Protect Record 80; Combine Table Output 72; Source Index 74; Zero Detect 90; Learned Address Logic 88 (as further described in FIG. 6 and including variations of the add key logic circuitry presented in FIGS. 7 and FIGS. 9–16, and including the symbol use count

[1] The parties dispute whether bracketed text of the claim element below is subject to 35 U.S.C. § 112 ¶ 6.

found in claim[s]: 8	a directory table.	logic circuit as described in FIG. 8); Learned Route Logic 94. FIG. 3: Source and Destination Index Tables 98; Address Record Memory 100; Arithmetic Computation 102; Source Protect Record 110. FIG 4: Multicast Record List 134; Arithmetic Code Compression 138; Integer 140; Truncate 142; Mod (n) Hashing 144; Index 136; Compressed Address Directory 130. FIG. 5: Index Table Memory 68; Arithmetic Computation Logic 72; Record Memory 78; Learned Address Logic 88; or, in the alternative, programmable devices implementing the logical functions of the dedicated circuitry of FIG. 5 as described above.; or, in the alternative, the host system of FIG. 5 performing some or all of the processing performed by the dedicated circuitry of FIG. 5 as described above.
means for filtering the data packet in response to the source filtering information found in claim[s]: 8	The means for filtering the data packet is at least the buffered routing logic (Figure 2 – item 56). The data is filtered in response to the source filtering information (as construed herein).	FIG. 2: the protect record 80, the learned route logic 94, and the buffered routing logic 56. FIG 3: Source Protect Record 110; Source and Destination Index Tables 98; Arithmetic Computation 102; Address Record Memory 100; Destination Route Record 112; Line 114. FIG 4: Multicast Record List 134. FIG 5: Index Table Memory 68; Arithmetic Computation Logic 72; Record Memory 78; Learned Address Logic 88; or, in the alternative, programmable devices implementing the logical functions of the dedicated circuitry of FIG. 5 as described above.; or, in the alternative, the host system of FIG. 5 performing some or all of the processing performed by the dedicated circuitry of FIG. 5 as described above.
means for looking up, using the destination address, in a routing table information associated with the	The means for looking up is at least the routing table index and the directory (Figure 4 – items 136, 130). The routing table index uses the destination address to look up information associated with the	Fig 2: Learned Address logic 88 (as further described in FIG. 6 and including variations of the add key logic circuitry presented in FIGS. 7 and FIGS. 9–16, and including the symbol use count logic

destination address for routing the data packet for delivery to the receiver found in claim[s]: 12	destination address.	circuit as described in FIG. 8); destination index table 66; combine table output 70; route index 76; destination routing table 84; route record 86; routing logic 56; zero detect 90; learned route logic 94. FIG 3: Source Protect Record 110; Source and Destination Index Tables 98; Arithmetic Computation 102; Address Record Memory 100; Destination Route Record 112; Line 114. FIG 4: Multicast Record List 134; Arithmetic Code Compression 138; Integer 140; Truncate 142; Mod (n) Hashing 144; Index 136; Compressed Address Directory 130. FIG 5: Index Table Memory 68; Arithmetic Computation Logic 72; Record Memory 78; Learned Address Logic 88; or, in the alternative, programmable devices implementing the logical functions of the dedicated circuitry of FIG. 5 as described above.; or, in the alternative, the host system of FIG. 5 performing some or all of the processing performed by the dedicated circuitry of FIG. 5 as described above.

III. P.R. 4-3(c): TIME NECESSARY FOR CLAIM CONSTRUCTION HEARING

The parties believe that two hours per side (four hours total) will be sufficient for the Claim Construction Hearing.

IV. P.R. 4-3(d): WITNESSES AT CLAIM CONSTRUCTION HEARING

Neither party identified any testimony of percipient and expert witnesses they contend support their respective claim constructions under P.R. 4-2(b). Pursuant to P.R. 4-3(d), neither party proposes to call any witness at Claim Construction Hearing.

V. P.R. 4-3(e): OTHER ISSUES

The Parties have not identified any other issues which might appropriately be taken up at a prehearing conference prior to the Claim Construction Hearing.

Dated: December 3, 2008

Respectfully submitted,

/s/ Trey Yarbrough

Trey Yarbrough
State Bar Number 22133500
Yarbrough & Wilcox PLLC
100 E. Ferguson, Ste. 1015
Tyler, Texas 75702
Telephone: (903) 595-3111
Facsimile: (903) 595-0191
E-mail: trey@yw-lawfirm.com

Attorneys for Defendant and Counter-Claimant
NETGEAR INC.,
ZYXEL COMMUNICATIONS, INC.,
D-LINK SYSTEMS, INC.,
and on behalf of Defendants:
3COM CORPORATION
FOUNDRY NETWORKS, INC.,
EXTREME NETWORKS, INC.,
NETGEAR INC.,
ZYXEL COMMUNICATIONS, INC.,
D-LINK SYSTEMS, INC.,
SMC NETWORKS, INC.,
TELLABS, INC.,
TELLABS NORTH AMERICA, INC., AND
ENTERASYS NETWORKS, INC.,

Of Counsel:

Steven H. Morrissett
Ming-Tao Yang
Aaron J. Capron
FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.
3300 Hillview Avenue
Palo Alto, CA 94304
Telephone: (650)849-6600
Facsimile: (650)849-6666
Email: morrissett@finnegan.com

E. Robert Yoches
FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.
901 New York Avenue, NW
Washington, DC 20001-4413
Telephone: (202)408-4000
Facsimile: (202)408-4400
Email bob.yoches@finnegan.com

/s/ Brett C. Govett
Robert M. Chiaviello, Jr.
Texas Bar No. 04190720
bobc@fulbright.com
Brett C. Govett
Texas Bar No. 08235900
bgovett@fulbright.com
FULBRIGHT & JAWORSKI L.L.P.
2200 Ross Avenue, Suite 2800
Dallas, TX 75201
Telephone: (214) 855-8000
Facsimile: (214) 855-8200

Attorneys for Plaintiff
FENNER INVESTMENTS, LTD.

OF COUNSEL

Otis W. Carroll
Texas Bar No. 03895700
IRELAND, CARROLL & KELLEY, P.C.
6101 South Broadway
P.O. Box 7879
Tyler, Texas 75711-7879
Telephone: (903) 561-1600
Facsimile: (903) 581-1071
Email: otiscarroll@icklaw.com

Franklin Jones, Jr.
Texas Bar No. 00000055
201 West Houston Street
P.O. Box 1249
Marshall, Texas 75671-1249
Telephone: (903) 938-4395
Facsimile: (903) 938-3360
Email: maizieh@millerfirm.com

T. John Ward, Jr.
Texas Bar No. 00794818
Law Office of T. John Ward, Jr.
P.O. Box 1231
109 W. Tyler
Longview, Texas 75601
Telephone: (903) 757-6400
Facsimile: (903) 757-2323
Email: jw@jwfirm.com

CERTIFICATE OF SERVICE

The undersigned certifies that the foregoing document was filed electronically in compliance with Local Rule CV-5(a). As such, this motion was served on all counsel who have consented to electronic service, Local Rule CV-5(a)(3)(A), on this the 3rd day of December, 2008.

/s/ Trey Yarbrough
Trey Yarbrough